

1 **CLAIMS**

2  
3 1. A method comprising:

4 processing an image using vector transformation followed by vector  
5 quantization to produce image values representative of the image;  
6 forming a bitstream containing the image values; and  
7 applying various protection levels to different parts of the bitstream.

8  
9 2. A method as recited in claim 1, wherein the processing comprises:

10 computing a vector transformation of the image to produce multiple  
11 vectors;

12 calculating a mean value of each of the vectors;

13 quantizing the mean values to form quantized values;

14 removing the mean values from corresponding vectors to produce  
15 normalized vectors; and

16 coding the normalized vectors using vector quantization and a bit allocation  
17 map to produce vector indexes.

18  
19 3. A method as recited in claim 1, wherein the vector transformation

20 comprises:

21 sub-sampling the image into sub-images;

22 dividing each of the sub-images into blocks of pixels;

23 transforming each of the blocks of pixels to produce coefficients; and

24 regrouping the coefficients of the sub-images into one or more vectors.  
25

1           4. A method as recited in claim 1, wherein the vector transformation  
2 forms vectors representative of the image, further comprising ordering the vectors  
3 in codebooks and assigning vector indexes to the vectors such that a bit error  
4 occurring at a less significant bit in a vector index results in less distortion than a  
5 bit error occurring at a more significant bit in the vector index.

6  
7           5. A method as recited in claim 4, wherein the ordering comprises:  
8 (a) selecting an initial vector;  
9 (b) assigning a vector index to the initial vector;  
10 (c) selecting a next vector that is nearest to the vector last assigned a vector  
11 index;  
12 (d) assigning a next vector index to the next vector; and  
13 (e) repeating the (c) selecting and the (d) assigning for all of the vectors.

14  
15           6. A method as recited in claim 4, wherein the applying comprises  
16 assigning different channel codes to different parts of the bitstream.

17  
18           7. A computer-readable medium comprising computer executable  
19 instructions that, when executed, direct a computing device to perform the method  
20 as recited in claim 1.

21  
22           8. A method comprising:  
23 ordering vectors representing a digital image in a codebook so that  
24 consecutive vectors are nearest neighbors to one another in n-dimensional space,  
25 where  $n > 1$ ;

1 assigning monotonically changing vector indexes to the vectors; and  
2 storing the codebook containing the vectors and the vector indexes in  
3 memory.

4  
5 **9.** A codebook data structure stored on a computer-readable medium  
6 that is produced as a result of the method of claim 8.

7  
8 **10.** An image coding system, comprising:  
9 a coder to process an image using vector transformation followed by vector  
10 quantization to produce image information, the coder forming a bitstream  
11 containing the image information; and  
12 an error protection coder to apply various levels of protection to different  
13 parts of the bitstream.

14  
15 **11.** An image coding system as recited in claim 10, wherein the coder is  
16 configured to compute the vector transformation by sub-sampling the image into  
17 sub-images, dividing each of the sub-images into blocks of pixels, transforming  
18 each of the blocks of pixels to produce coefficients, and regrouping the  
19 coefficients of the sub-images into one or more vectors.

1           **12.**    An image coding system as recited in claim 10, wherein the vector  
2 transformation forms vectors representative of the image, the coder further  
3 comprising a codebook generator to generate an ordered codebook that contains  
4 the vectors and corresponding vector indexes that reference the vectors, the  
5 ordered codebook arranging the vectors such that a bit error occurring at a less  
6 significant bit in a vector index results in less distortion than a bit error occurring  
7 at a more significant bit.

8  
9           **13.**    An image coding system as recited in claim 12, wherein the  
10 codebook generator organizes the codebook by ordering the vectors such that a  
11 next vector is nearest to a previous vector and assigning monotonically changing  
12 vector indexes to the vectors.

13  
14           **14.**    An image coding system as recited in claim 10, wherein the error  
15 protection coder assigns different channel codes to different parts of the bitstream.

16  
17           **15.**    An image coding system as recited in claim 10, embodied as  
18 computer-readable instructions stored on a computer-readable medium.

19  
20           **16.**    An image distribution architecture, comprising:  
21           a source having an image encoder to transform an image into multiple  
22 vectors and quantize the vectors to produce image values, the image encoder  
23 ordering the vectors in codebooks and assigning vector indexes to the vectors such  
24 that a bit error occurring at a less significant bit in a vector index results in less  
25 distortion than a bit error occurring at a more significant bit;

1 the source further having a UEP (Unequal Error Protection) coder that  
2 layers the vector indexes according to significance, the source transmitting a bit  
3 stream including at least the image values and the layered vector indexes over a  
4 communication channel; and

5 a destination receiving the bitstream, the destination having a decoder to  
6 recover the vectors using the vector indexes and reconstruct the image from the  
7 image values and the vectors.

8  
9 **17.** An image distribution architecture as recited in claim 16, wherein  
10 the image encoder is configured to allocate different numbers of bits to different  
11 groups of the vectors.

12  
13 **18.** An image distribution architecture as recited in claim 16, wherein  
14 the UEP coder assigns different channel codes to layers of the vector indexes.

15  
16 **19.** A bitstream data structure stored on a computer-readable medium,  
17 comprising:

18 a first field containing mean values calculated from vectors produced from  
19 a digital image;

20 a second field containing a bit allocation map that allocates bits to various  
21 groups of the vectors; and

22 a third field containing vector indexes used to reference the vectors in a  
23 codebook, the vector indexes being layered.